



ATF Tools V0.2

User Guide

Contents

1. Introduction	2
2. png2atf	3
3. pvr2atf	5
4. atf2pvr	6
5. ATFViewer	7
6. atfinfo	7
7. pngalpha	8
8. pngsquare	8

1. Introduction

This document is a reference for the tools which are made available to create and manage ATF files (Adobe Texture Files). ATF files are the recommended file type for fixed texture assets when used in conjunction with the Flash Player 'Molehill' APIs.

1.1 ATF basics

ATF files are primarily a file container to store lossy texture data. It achieves its lossy compression through the use of two common techniques: JPEG-XR¹ compression and block based compression. JPEG-XR compression provides a competitive method to save storage space and network bandwidth. Block based compression provides a way to reduce texture memory usage on the client, at a fixed ratio of 1:8 compared to RGBA textures. ATF supports three types of block based compression: DXT1², ETC1³ and PVRTC⁴.

In ATF files compression is performed on two levels: first an optional block based compression and on top of that standard loss less or lossy JPEG-XR compression. Other features in ATF files include:

- Embedding of mip maps
- Optionally embeds a complete cube map (sky map)
- Optionally supports selection of internal color space (4:4:4, 4:2:2 and 4:2:0).

1.2 ATF limitations

ATF files have various limitations which are the direct result of existing hardware capabilities on various mobile devices. The specific limitations include:

- Texture sizes are limited to a maximum of 2048x2048 pixels.
- Texture sizes are limited to power of two numbers on each side, i.e. 1,2,4,8,16,32,64,128,256,512,1024 and 2048.
- Block based compression is limited to square textures, i.e. 1x1,2x2,4x4,8x8,16x16,32x32,64x64,128x128,256x256,512x512,1024x1024 and 2048x2048.
- Block based compression does not support an alpha channel (transparency) of any kind at this time.
- At it's highest resolution of 2048x2048 you are required to provide at least one regular mip map level because certain devices only allow texture sizes of up to 1024x1024. Molehill will automatically switch to 1024x1024 if it sees a need for it.

1 <http://www.itu.int/rec/T-REC-T.832>

2 <http://msdn.microsoft.com/en-us/library/bb147243%28v=VS.85%29.aspx>

3 http://www.khronos.org/registry/gles/extensions/OES/OES_compressed_ETC1_RGB8_texture.txt

4 <http://www.imgtec.com/powervr/insider/powervr-pvrtexlib.asp>

2. png2atf

png2atf is a command line utility which converts PNG files to ATF files. The resulting ATF files can then be directly used with the `uploadCompressedTextureFromByteArray()` ActionScript 3 API. **png2atf** takes any valid PNG file and by default converts it to either a RGB or RGBA ATF file, depending on if the PNG file has transparency. It can optionally also create a block based compression texture if the source PNG has no transparency.

2.1 Invocation

To convert a PNG file to a RGB or RGBA ATF file run the command as such:

```
C:\> png2atf -i test.png -o test.atf
[In 4096KB][Out 410KB][Ratio 10.0241%][LZMA:0KB JPEG-XR:410KB]
```

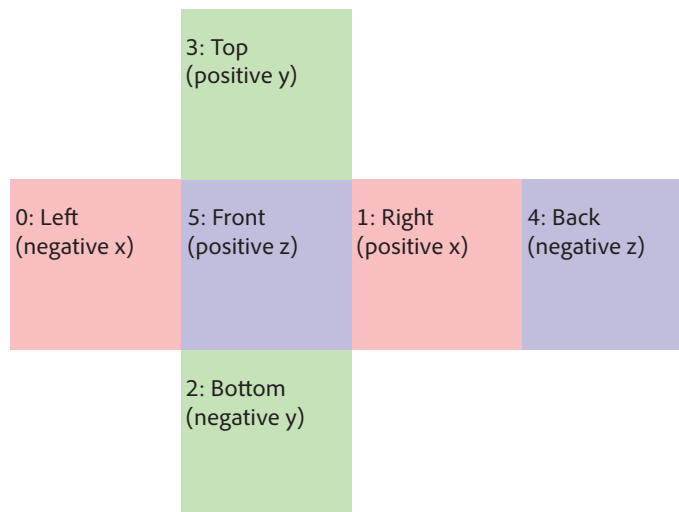
To create a block based compression texture file run the command as such:

```
C:\> png2atf -c -i test.png -o test.atf
.....
[In 2048KB][Out 1704KB][Ratio 83.2007%][LZMA:937KB JPEG-XR:766KB]
```

2.2 Command line options

- i <file> specifies the input file name
- o <file> specifies the output file name
- c creates a block based compression texture. Three types of block based compression will be used and embedded into the same ATF file: DXT1, ETC1 and PVRTC 4bpp. When you load this type of texture into the Flash Player, the Flash Player will pick the appropriate format for the device.

If the PNG file has transparency a RGBA texture will be generated instead. You can use the **atfinfo** tool later to find out what format was chosen or you can use the **pngalpha** tool to find out if a particular PNG file has transparent pixels before you use this option.
- m create a cube map texture. Input files need to be named consecutively in this form: XXXn.png where XXX is the base file name and n=0-5. Example: 'cube0.png', 'cube1.png', 'cube2.png', 'cube3.png', 'cube4.png', 'cube5.png'. The input file name in this case should be 'cube0.png'.



- n By default **png2atf** will auto generate all applicable mip map levels. In some cases it is not desirable to enable mip maps like for instance for sky maps; you can use this option to turn off the auto generation of mip maps.
- s Silences any output the tool might have during compression. This can be useful for batch processing.
- 4 Instructs the JPEG-XR encoder to use a 4:4:4 color space internally. This is the default for block based compression. In some cases it is desirable to use this color space for RGB/RGBA textures in case you see color bleeding artifacts around red or blue shapes or for normal maps.
- 2 Instructs the JPEG-XR encoder to use a 4:2:2 color space internally. It is not recommended to use this color space for block based compression as this can cause severe artifacts.
- 0 Instructs the JPEG-XR encoder to use a 4:2:0 color space internally. This is the default for RGB/RGBA textures. It is not recommended to use this color space for block based compression as this can cause severe artifacts; though it might be worth experimenting with this option if for instance the image data is monochromatic.
- q <0-180> Selects the quantization level, i.e. how much the image data should be compressed. A value of 0 means loss less compression which is the default for block based compression textures. It is not recommended to use lossy compression for block based compression; though it is worth to experiment with this value from a case to case basis.

The default value for RGB and RGBA textures is 30. Higher values will create more artifacts, smaller values will reduce artifacts. Note that you should not confuse this with the standard JPEG compression factor. It's similar in concept but applies in a different non-linear range.
- f <0-15> Selects how many flex bits should be trimmed during JPEG-XR compression. This option is not related to the quantization level but selects how much noise should be retained across the image. Like the quantization level higher values create more artifact. The default value is always 0.

3. pvr2atf

pvr2atf is a command line utility which converts PVR texture files to ATF files which can then be used with the `uploadCompressedTextureFromByteArray()` ActionScript 3 API. The tool works similarly to **png2atf** except that you have to provide input files in the PVR texture format. For block based compression you have to provide already compressed textures in three PVR files.

3.1 Invocation

To convert a PVR file to a RGB or RGBA ATF file run the command as such:

```
C:\> png2atf -i test.pvr -o test.atf
.
[In 4096KB][Out 410KB][Ratio 10.0241%][LZMA:0KB JPEG-XR:410KB]
```

To create a block based compression texture file run the command as such, where each input PVR file represents a block compressed file format:

```
C:\> png2atf -d test_dxt1.pvr -e test_etc1.pvr -p test_pvrtc.pvr -o test.atf
.....
[In 2048KB][Out 1704KB][Ratio 83.2007%][LZMA:937KB JPEG-XR:766KB]
```

3.2 Accepted PVR texture types

pvr2atf accepts the following types of PVR files:

- OpenGL ES 2.0 RGB 888 (OGL_RGB_888)
- OpenGL ES 2.0 RGBA 8888 (OGL_RGBA_8888)
- DirectX 9 DXT 1 (D3D_DXT1)
- OpenGL ES 2.0 ETC (ETC_RGB_4BPP)
- OpenGL ES 2.0 PVRTC 4BPP (OGL_PVRTC4)
- Cube maps (PVRTEX_CUBEMAP)
- Mip maps (PVRTEX_MIPMAP)

pvr2atf does not accept flipped textures. Make sure that you create PVR texture with the flipping option turned off. This can be achieved with the '-yflip0' option of the PVR TexTool⁵ command line tool or by unchecking the 'Flipped' checkbox in the PVR TexTool GUI tool.

3.2 Command line options

- | | |
|-----------|--|
| -r <file> | specifies the input RGB or RGBA PVR file. Note that you can't mix this with the -d/-e/-p options. |
| -d <file> | specifies the input DXT1 PVR file. |
| -e <file> | specifies the input ETC1 PVR file |
| -p <file> | specifies the input PVRTC PVR file. |
| -o <file> | specified the output file name |
| -s | Silences any output the tool might have during compression. This can be useful for batch processing. |
| -4 | Instructs the JPEG-XR encoder to use a 4:4:4 color space internally. This is the default |

5 <http://www.imgtec.com/powervr/insider/powervr-pvrtextrtool.asp>

for block based compression. In some cases it is desirable to use this color space for RGB/RGBA textures in case you see color bleeding artifacts around red or blue shapes or for normal maps.

- 2 Instructs the JPEG-XR encoder to use a 4:2:2 color space internally. It is not recommended to use this color space for block based compression as this can cause severe artifacts.
- 0 Instructs the JPEG-XR encoder to use a 4:2:0 color space internally. This is the default for RGB/RGBA textures. It is not recommended to use this color space for block based compression as this can cause severe artifacts; though it might be worth experimenting with this option if for instance the image data is monochromatic.
- q <0-180> Selects the quantization level, i.e. how much the image data should be compressed. A value of 0 means loss less compression which is the default for block based compression textures. It is not recommended to use lossy compression for block based compression; though it is worth to experiment with this value from a case to case basis.

The default value for RGB and RGBA textures is 30. Higher values will create more artifacts, smaller values will reduce artifacts. Note that you should not confuse this with the standard JPEG compression factor. It's similar in concept but applies in a different non-linear range.
- f <0-15> Selects how many flex bits should be trimmed during JPEG-XR compression. This option is not related to the quantization level but selects how much noise should be retained across the image. Like the quantization level higher values create more artifact. The default value is always 0.

4. atf2pvr

atf2pvr converts/splits ATF files into PVR texture files. If you specify an ATF file with block based compression, three PVR files will be generated with distinct file extensions to identify the type.

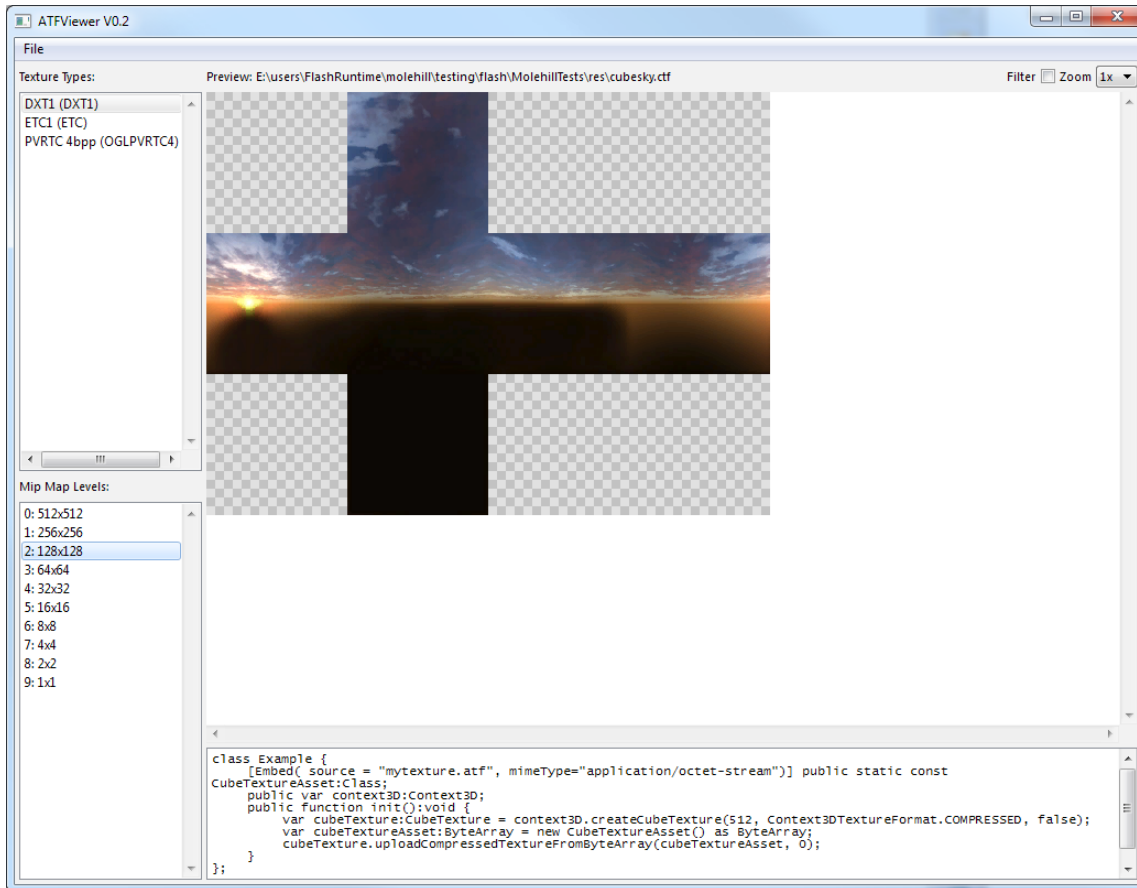
4.1 Invocation

```
C:\> atf2pvr -i test.atf -o test
```

5. ATFViewer

ATFViewer is a GUI tool which previews and inspects ATF files. The primary purpose is to audit DXT1, ETC1 and PVRTC compression artifacts. You can open and view ATF files by either using the 'Open...' menu item or by dragging a file from Explorer into the window.

The Snippet preview area shows you an example of how to load a particular ATF file in ActionScript 3. code.



6. atfinfo

atfinfo is command line utility which displays internal information about ATF files. It print size, mip map count, texture type, texture format and whether it is a cube map. It also shows which ActionScript 3 classes and format correspond to a particular format.

6.1 Invocation

```
C:\> atfinfo -i test.atf
File Name           : test.atf
ATF File Type       : Compressed (DXT1+ETC1+PVRTC4bpp)
Size                : 1024x1024
Cube Map            : no
Mip Map Count       : 10 (512x512,256x256,128x128,64x64,32x32,16x16,8x8,4x4,2x2,1x1)
AS3 Texture Class   : Texture (flash.display3D.Texture)
AS3 Texture Format   : Context3DTextureFormat.COMPRESSED (flash.display3D.Context3DTextureFormat)
```

7. pngalpha

pngalpha is a command line utility which determines whether a PNG file has transparent pixels or not. It simply prints out 'alpha' if the PNG has transparent pixels, 'opaque' if it does not have transparent pixels.

7.1 Invocation

```
C:\> pngalpha -i test.png  
alpha
```

8. pngsquare

pingsquare is a command line utility which determines whether a PNG file is square or rectangular. It simply prints out 'square' if the PNG is square, 'rectangle' if the PNG is not square.

8.1 Invocation

```
C:\> pngsquare -i test.png  
square
```